

The image shows a 64x64 grid of binary symbols, likely representing the state of a cellular automaton. The symbols are arranged in a repeating pattern of four columns. The first column contains 'SSS' symbols. The second column contains 'SSSS' symbols. The third column contains 'SSSSS' symbols. The fourth column contains 'SSSSSS' symbols. The symbols are arranged in a staggered, wave-like pattern across the grid.

SSSSSSSS YY YY SSSSSSSS LL KK KK WW WW SSSSSSSS EEEEEEEEEE TTTTTTTTTT
SSSSSSSS YY YY SSSSSSSS LL KK KK WW WW SS SSSSSSSS EEEEEEEEEE TTTTTTTTTT
SS YY YY SS LL KK KK WW WW SS SSSSSSSS EEE EEE TTT TTT
SS YY YY SS LL KK KK WW WW SS SSSSSSSS EEE EEE TTT TTT
SSSSSS SSSSSS SS LL KKKKKK WW WW SSSSSSS EEE EEE TTT TTT
SS YY YY SS LL KK KK WW WW SS SSSSSSS EEE EEE TTT TTT
SS YY YY SS LL KK KK WWWWWWWWWWW SS SSSSSSS EEE EEE TTT TTT
SS YY SSSSSSSS LLLLLLLL KK KK WW WW SSSSSSS EEE EEE TTT TTT
SS YY SSSSSSSS LLLLLLLL KK KK WW WW SSSSSSS EEE EEE TTT TTT
LL IIIII SSSSSSSS
LL IIIII SS
LL IIIII SS
LL IIIII SS
LL IIIII SSSSSS
LL IIIII SSSSSS
LL IIIII SS
LL IIIII SS
LL IIIII SS
LL IIIII SSSSSS
LL IIIII SSSSSS

(2)	69	DECLARATIONS
(3)	116	LOCK/UNLOCK PAGES IN WORKING SET/MEMORY
(4)	255	UNLOCK GLOBAL PAGE FROM WORKING SET
(5)	292	LOCK/UNLOCK SINGLE PAGE IN WORKING SET/MEMORY
(7)	567	SWAPLOCK - INTERFACE TO SWAPWSLE FROM LOCK/UNLOCK
(8)	597	SWAPWSLE - SWAP WORKING SET LIST ENTRIES
(10)	695	SCNWSLX - SCAN FOR WORKING SET LIST INDEX
(12)	770	MMGSEXPKSTACK - EXPAND THE KERNEL STACK

0000 1 .TITLE SYSLKWSET - LOCK/UNLOCK FROM WORKING SET/MEMORY
0000 2 .IDENT 'V04-000'
0000 3 .
0000 4 *****
0000 5 *
0000 6 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 * ALL RIGHTS RESERVED.
0000 9 *
0000 10 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 * TRANSFERRED.
0000 16 *
0000 17 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 * CORPORATION.
0000 20 *
0000 21 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23 *
0000 24 *
0000 25 *****
0000 26 *
0000 27 ++
0000 28 :FACILITY:
0000 29 :
0000 30 :ABSTRACT:
0000 31 :
0000 32 :ENVIRONMENT:
0000 33 :
0000 34 :AUTHOR: PETER H. LIPMAN , CREATION DATE: 24-APR-78
0000 35 :
0000 36 :MODIFIED BY:
0000 37 :
0000 38 : V03-009 WMC0005 Wayne Cardoza 26-Feb-1984
0000 39 : Support for resident global sections.
0000 40 :
0000 41 : V03-008 WMC0004 Wayne Cardoza 28-Nov-1983
0000 42 : Move kernel stack expansion WSL slots.
0000 43 :
0000 44 : V03-007 WMC0003 Wayne Cardoza 13-Oct-1983
0000 45 : Add support for expanding kernel stack.
0000 46 :
0000 47 : V03-006 LJK0247 Lawrence J. Kenah 1-Sep-1983
0000 48 : Remove fatal bugcheck when MMG\$SCNWSLX fails. The global
0000 49 : page was removed from the working set and should simply be
0000 50 : faulted another time.
0000 51 :
0000 52 : V03-005 TCM0001 Trudy C. Matthews 1-Apr-1983
0000 53 : Change references to working set fields in PHD so that
0000 54 : they are used as unsigned words.
0000 55 :
0000 56 : V03-004 WMC0002 Wayne Cardoza 02-Mar-1983
0000 57 : MMG\$CRECOM1 HAS GONE AWAY

SYSLKWSET
V04-000

- LOCK/UNLOCK FROM WORKING SET/MEMORY

J 3

16-SEP-1984 02:21:29 VAX/VMS Macro V04-00
5-SEP-1984 03:54:52 [SYS.SRC]SYSLKWSET.MAR;1

Page 2
(1)

0000 58 :
0000 59 :
0000 60 :
0000 61 :
0000 62 :
0000 63 :
0000 64 :
0000 65 :
0000 66 :
0000 67 ;--

V03-003 WMC0001 Wayne Cardoza 29-Sep-1982
Fix loop when trying to reduce working set to quota.
V03-002 KDM46395 Kathleen D. Morse 28-Jun-1982
Check whether page table pages should be moved out
of the locked portion of the working set when the
PTWSLELCK byte array count goes to zero.

0000 69 .SBTTL DECLARATIONS
0000 70 ;
0000 71 : INCLUDE FILES:
0000 72 :
0000 73 : \$IPLDEF
0000 74 : \$MMGDEF
0000 75 : \$OPDEF
0000 76 : \$PCBDEF
0000 77 : \$PFNDEF
0000 78 : \$PHDDEF
0000 79 : \$PRDEF
0000 80 : \$PRVDEF
0000 81 : \$PTEDEF
0000 82 : \$SSDEF
0000 83 : \$VADEF
0000 84 : \$WQHDEF
0000 85 : \$WSLDEF
0000 86 :
0000 87 : EXTERNAL SYMBOLS:
0000 88 :
0000 89 :
0000 90 : MACROS:
0000 91 :
0000 92 :
0000 93 : EQUATED SYMBOLS:
0000 94 :
0000 95 : OFFSET FROM AP
0000 96 :
0000 97 : INADR = 4 :OFFSET TO INPUT RANGE
0000 98 : RETADR = 8 :OFFSET TO RETURN RANGE
0000 99 : ACMODE = 12 :ACCESS MODE
0000 100 :
0000 101 : BIT DEFINITIONS IN R8 CONTROL LONG WORD
0000 102 :
0000 103 : _VIELD LCK,8,<-
0000 104 : <LCKPAG,,M>,- :SET IF LOCKING, CLEAR IF UNLOCKING
0000 105 : <GLOBAL,,M> - :SET IF GLOBAL PAGE
0000 106 : >
0000 107 :
0000 108 : OWN STORAGE:
0000 109 :
0000 110 : .PSECT \$\$S210, LONG
0000 111 :
0000 112 : MMG\$GL_PFNLOCK::
0000 113 : .LONG 1@30 :DOWN COUNTER OF PAGES REMAINING
0004 114 : THAT MAY BE LOCKED IN MEMORY

0004 116 .SBTTL LOCK/UNLOCK PAGES IN WORKING SET/MEMORY
0004 117 ++
0004 118 FUNCTIONAL DESCRIPTION:
0004 120 THE LKWSET SYSTEM SERVICE LOCKS THE SPECIFIED RANGE OF PAGES
0004 121 INTO THE WORKING SET. THE PAGES LOCKED ARE NO LONGER CANDIDATES
0004 122 FOR PAGE REPLACEMENT AND ARE THUS GUARANTEED RESIDENT IN MEMORY WHEN
0004 123 THE PROCESS IS IN THE BALANCE SET.
0004 124 SINCE SOME REASONABLE NUMBER OF DYNAMIC PAGES IS REQUIRED IN ORDER
0004 125 TO GUARANTEE THAT AN INSTRUCTION CAN RUN TO COMPLETION, THE NUMBER OF
0004 126 PAGES THAT MAY BE LOCKED IN THE WORKING SET IS LIMITED.
0004 127 THE ULWSET SYSTEM SERVICE UNLOCK THE PAGES FROM THE WORKING SET.
0004 128
0004 129 CALLING SEQUENCE:
0004 130
0004 131 CALLG ARGLIST,G^SYSSLKWSET
0004 132 CALLG ARGLIST,G^SYSSULWSET
0004 133 CALLG ARGLIST,G^SYSSLCKPAG
0004 134 CALLG ARGLIST,G^SYSSULKPAG
0004 135
0004 136 INPUT PARAMETERS:
0004 137
0004 138 INADR(AP) = ADDRESS OF 2 LONG WORDS THE 1ST OF WHICH SPECIFIES
0004 139 THE STARTING VIRTUAL ADDRESS TO LOCK/UNLOCK, THE 2ND SPECIFIES
0004 140 THE ENDING VIRTUAL ADDRESS TO LOCK/UNLOCK (INCLUSIVE).
0004 141 RETADR(AP) = ADDRESS OF A 2 LONGWORD ARRAY INTO WHICH IS RETURNED
0004 142 THE STARTING AND ENDING VIRTUAL ADDRESSES (INCLUSIVE)
0004 143 OF THE PAGES JUST LOCKED/UNLOCKED
0004 144 ACMODE(AP) = THE ACCESS MODE (MAXIMIZED WITH CALLING MODE)
0004 145 ON BEHALF OF WHICH THE LOCK/UNLOCK IS TO BE DONE
0004 146 THIS MODE MUST BE AT LEAST AS PRIVILEGED AS THE MODE
0004 147 OF THE PAGE OWNER.
0004 148
0004 149 IMPLICIT INPUTS:
0004 150
0004 151 NONE
0004 152
0004 153 OUTPUT PARAMETERS:
0004 154 R0 = SYSTEM STATUS CODE
0004 155
0004 156
0004 157 IMPLICIT OUTPUTS:
0004 158
0004 159 NONE
0004 160
0004 161 COMPLETION CODES:
0004 162
0004 163 SSS_WASSET :PREVIOUS STATE OF LOCK WAS SET
0004 164 SSS_WASCLR :PREVIOUS STATE OF LOCK WAS CLEAR
0004 165 SSS_ACCVIO :ACCESS VIOLATION
0004 166 SSS_NOPRIV :NO PRIVILEGE TO LOCK/UNLOCK PAGE
0004 167 SSS_LKWSETFUL :LOCKED PORTION OF WORKING SET IS FULL
0004 168 SSS_LCKPAGFUL :NO MORE PAGES MAY BE LOCKED IN MEMORY
0004 169 SSS_PAGOWNVIO :PAGE OWNER VIOLATION
0004 170
0004 171 SIDE EFFECTS:
0004 172 ;

```

0004 173 : NONE
0004 174 :
0004 175 :--
0004 176 :
0004 177 :
0004 178 : *****
0004 179 : ***** THE FOLLOWING CODE MAY BE PAGED *****
0004 180 : *****
0004 181 : *****
00000000 182 : .PSECT Y$EXEPAGED
0000 183 :
0000 184 :
0000 185 :
0000 186 : .ENABL LSB
0000 187 :
01FC 0000 188 : .ENTRY EXE$ULWSET,^M<R2,R3,R4,R5,R6,R7,R8>
0002 189 :
58 20 9A 0002 190 : MOVZBL #WSL$M_WSLOCK,R8 ;UNLOCKING FROM WORKING SET
05 11 0005 191 : BRB 10$ 
0007 192 :
01FC 0007 193 : .ENTRY EXE$ULKPAG,^M<R2,R3,R4,R5,R6,R7,R8>
0009 194 :
58 10 9A 0009 195 : MOVZBL #WSL$M_PFNLOCK,R8 ;UNLOCKING FROM MEMORY
50 09 3C 000C 196 10$: MOVZWL #SSS_WASSET,RO ;INIT TO 'PAGES WERE LOCKED'
5E 1C C2 000F 197 : SUBL #-MMGSC_LENGTH,SP ;RESERVE SCRATCH AREA
27 11 0012 198 : SAVIPL 30$ ;SAVE CALLERS IPL FOR EXIT
27 11 0015 199 : BRB 30$ 
01FC 0017 200 : .ENTRY EXE$LKWSET,^M<R2,R3,R4,R5,R6,R7,R8>
0019 201 :
58 20 9A 0019 203 : MOVZBL #WSL$M_WSLOCK,R8 ;LOCKING IN WORKING SET
0E 11 001C 204 : BRB 20$ 
001E 205 :
01FC 001E 206 : .ENTRY EXE$LCKPAG,^M<R2,R3,R4,R5,R6,R7,R8>
0020 207 :
50 24 3C 0020 208 : MOVZWL #SSS_NOPRIV,RO ;ASSUME CANNOT LOCK PAGES IN MEMORY
0023 209 : IFNPRIV PSWAPM,45$ ;BRANCH IF CANNOT LOCK PAGES IN MEMORY?
58 10 9A 0029 210 : MOVZBL #WSL$M_PFNLOCK,R8 ;LOCKING IN MEMORY
5E 1C C2 002C 211 20$: SUBL #-MMGSC_LENGTH,SP ;RESERVE SCRATCH AREA
41 10 0035 212 : DSBINT #IPLS_A$TDEL ;PREVENT SYSTEM WORKING SET GROWTH AST
50 01 3C 0037 213 : BSBB 50$ ;SET QUOTA WORKING SET SIZE BEFORE LOCK
00 58 08 E2 003A 214 25$: MOVZWL #SSS_WASCLR,RO ;INIT TO 'PAGES WERE UNLOCKED'
56 00000011'EF DE 003E 215 : BBSS #LCK_V_LCKPAG,R8,30$ ;SET LOCK FLAG
58 10 50 F0 0045 216 30$: MOVAL MMGS$[CRULKPAG,R6 ;ADDRESS OF PAGE LOCK/UNLOCK ROUTINE
FFB3' 30 004A 218 : INSV RO,#16,#16,R8 ;INITIAL STATUS IN HIGH 16 BITS
27 50 E9 004D 219 : BSBW MMGSINADRINI ;INITIALIZE RETURN ADDRESS ARAYS & SCRATCH A
52 54 7D 0050 220 : BLBC R0,45$ ;R2 = START OF RANGE, R3 = END
FFAA' 30 0053 221 : MOVQ R4,R2 ;COMMON CREATE CODE
50 DD 0056 222 : BSBW MMGSCREDEL ;SAVE STATUS
FFA5' 30 0058 223 : PUSHL R0 ;RETURN AFFECTED ADDRESS RANGE
19 50 E9 005B 224 : BSBW MMGSRETRANGE ;USE THIS BAD STATUS RATHER THAN CREDEL
50 BA 005E 225 : BLBC R0,45$ 
50 E8 0060 226 : POPR R0
11 50 E8 0063 227 : BLBS R0,40$ ;BRANCH IF SUCCESSFUL
50 0194 8F B1 0063 227 : CMPW #SSS_LKWSETFUL,RO ;IS THIS LOCKED WORKING SET FULL STATUS
0A 12 0068 228 : BNEQ 40$ ;BRANCH IF NOT
OC 10 006A 229 : BSBB 50$ ;REVERIFY SYSTEM DIDN'T SHRINK LIMIT

```

50	C8 52	E8 006C	230	BLBS	R2,25\$	
	0194 8F	3C 006F	231	MOVZWL	#SSS_LKWSETFUL,RO	;BRANCH IF ADJUSTMENT OCCURED
		0074	232	ENBINT		;RESET ERROR RETURN STATUS
		04 0077	233	40\$: RET		;RESTORE CALLERS IPL
55	00000000'GF	D0 0078	235	MOVL	G^CTL\$GL_PHD,R5	;GET PROCESS HEADER ADDRESS
50	18 A5 08 A5	D4 007F	236	CLRL	R2	;ASSUME NO ADJUSTMENT RETURN STATUS
	50	A3 0081	237	SUBW3	PHDSW_WSLIST(R5),PHDSW_WSQUOTA(R5),RO ;GET MAX LOCKABLE SIZE	
	50	B6 0087	238	INCW	RO	;IN ORIGIN 1
7E	50 A5	3C 0089	239	MOVZWL	RO,RO	;GET IT IN A LONGWORD
	50	3C 008C	240	MOVZWL	PHDSW_WSSIZE(R5),-(SP)	;GET CURRENT SIZE
	8E	C2 0090	241	SUBL	(SP)+,RO	;GET DIFFERENCE FROM CURRENT SIZE
7E	50 A5	B0 0095	242	BEQL	60\$;BRANCH IF CURRENT SIZE IS QUOTA, R0=0
	7E	D4 0099	243	MOVW	PHDSW_WSSIZE(R5),-(SP)	;SAVE STARTING SIZE
	50	32 009B	244	CLRL	-(SP)	;NO RETURN SIZE NEEDED
00000000'EF	02	FB 009E	245	CVTWL	RO,-(SP)	;FORM ADJUSTMENT SIZE
8E	50 A5	B1 00A5	246	CALLS	#2,SYSSADJWSL	;ADJUST THE WORKING SET SIZE
	05	13 00A9	247	CMPW	PHDSW_WSSIZE(R5),(SP)+	;DID SIZE CHANGE
52	01	88 00AB	248	BEQL	60\$;NO - GIVE UP
	D1	11 00AE	249	BISB	#1,R2	;SET ADJUSTMENT PERFORMED
	05	00B0	250	BRB	55\$;DO THIS TILL SIZE AT AUTHORIZED QUOTA
	00B1	251	60\$: RSB			;R0 LOW BIT INDICATES CHANGE OCCURRED
	00B1	252				
	00B1	253		.DSABL	LSB	

00B1 255 .SBTTL UNLOCK GLOBAL PAGE FROM WORKING SET
 00B1 256
 00B1 257 CALLING SEQUENCE:
 00B1 258
 00B1 259 BSBW MMGSULKGBLWSLE
 00B1 260
 00B1 261 INPUTS:
 00B1 262
 00B1 263 IPL = SYNCH
 00B1 264 R1 = WORKING SET LIST INDEX OF ENTRY TO UNLOCK (HIGH 16 BITS = 0)
 00B1 265 R2 = WORKING SET LIST ENTRY (VA AND PAGE TYPE)
 00B1 266 R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY
 00B1 267 R4 = PROCESS CONTROL BLOCK ADDRESS
 00B1 268 R5 = PROCESS HEADER ADDRESS (P1 SPACE OK)
 00B1 269
 00B1 270 OUTPUTS:
 00B1 271
 00B1 272 R2 PRESERVED
 00B1 273 R8 IS ALTERED
 00B1 274 PHD\$W_WSDYN(R5) = NEW WORKING SET LIST INDEX FOR THE UNLOCKED PAGE
 00B1 275
 00B1 276 *****
 00B1 277 ***** THE FOLLOWING CODE MUST BE RESIDENT *****
 00B1 278
 00B1 279
 00000000 280 .PSECT \$MMGCOD
 0000 281
 0000 282
 0000 283
 0000 284 MMGSULKGBLWSLE:::
 7E 08 DD 0000 285 PUSHL #IPL\$ SYNCH ;SAVE IPL FOR ENBINT TO RESTORE
 50 51 7D 0002 286 MOVQ R1 -(5P) ;PUSHL VA, PUSHL WSLX
 50 63 DD 0005 287 MOVL (R3), R0 ;GET PAGE TABLE ENTRY
 43 000F 0008 288 MOVL #<SS\$_WASSET @16 ! - ;SET UNLOCK PARAMETER
 43 11 000F 289 WSL\$M_WSLOCK ! WSL\$M_PFNLOCK>, R8 ;IN R8
 43 11 000F 290 BRB ULKGBLWSLE ;JOIN COMMON LOCK/UNLOCK CODE

58 00090030 8F 08 DD 0000
 50 63 DD 0002
 43 11 000F 0005
 43 11 000F 0008
 43 11 000F 000F
 43 11 000F 290

0011 292 .SBTTL LOCK/UNLOCK SINGLE PAGE IN WORKING SET/MEMORY
0011 293 ++
0011 294 FUNCTIONAL DESCRIPTION:
0011 295
0011 296
0011 297 CALLING SEQUENCE:
0011 298 BSBW MMG\$LCKULKPAG
0011 300
0011 301 INPUT PARAMETERS:
0011 302
0011 303 R0 = ACCESS MODE TO CHECK AGAINST PAGE OWNER
0011 304 R2 = VIRTUAL ADDRESS
0011 305 R4 = PCB ADDRESS
0011 306 R5 = PROCESS HEADER ADDRESS - P1 OR SYSTEM SPACE
0011 307 R6 = COUNT - 1 OF PAGES REMAINING TO BE DELETED INCLUDING THIS ONE
0011 308 R7 = +^X200 IF GOING FORWARD IN ADDRESS SPACE
0011 309 = -^X200 IF GOING BACKWARDS IN ADDRESS SPACE
0011 310 R8<0:7> = WSL\$M_WSLOCK IF LOCKING IN WORKING SET
0011 311 = WSL\$M_PFNLOCK IF LOCKING IN PHYSICAL MEMORY
0011 312 = EITHER OR BOTH OF THE ABOVE IF UNLOCKING
0011 313 R8<8>
0011 314 R8<16:31> = LCK_V_LCKPAG SET IF LOCKING, CLEAR IF UNLOCKING
0011 315 = SSS_WASSET OR SSS_WASCLR
0011 316
0011 317 IMPLICIT INPUTS:
0011 318 NONE
0011 319
0011 320 OUTPUT PARAMETERS:
0011 321
0011 322 R0 = SYSTEM STATUS CODE (INCLUDING 'WAS SET' BIT IF SUCCESSFUL)
0011 323 R2 - PRESERVED
0011 324 R8<16:31> = SSS_WASSET IF LOCKING AND THE LOCK HAD ALREADY BEEN SET
0011 325 = SSS_WASCLR IF UNLOCKING AND THE LOCK HAD ALREADY BEEN CLEAR
0011 326 - OTHERWISE IT IS PRESERVED.
0011 327
0011 328 IMPLICIT OUTPUTS:
0011 329 NONE
0011 330
0011 331 COMPLETION CODES:
0011 332
0011 333 SSS_WASSET :IF LOCKING, WAS ALREADY LOCKED
0011 334 SSS_WASCLR :IF UNLOCKING, WAS ALREADY UNLOCKED
0011 335 SSS_ACCVIO :ACCESS VIOLATION
0011 336 SSS_PAGOWNVIO :PAGE OWNER VIOLATION
0011 337 SSS_LKWSETFUL :IF LOCKING, LOCKED PORTION OF WORK SET FULL
0011 338
0011 339
0011 340 SIDE EFFECTS:
0011 341 NONE
0011 342
0011 343
0011 344 --

```

0011 346 : ****
0011 347 : **** THE FOLLOWING CODE MUST BE RESIDENT ****
0011 348 :
0011 349 :
0011 350 :
00000011 351 .PSECT $MMGCOD
0011 352 :
0011 353 :
0011 354 :
0011 355 .ENABL LSB
0011 356 :
0011 357 MMG$LCKULKPAG:::
      51   D4 0011 358 CLRL R1          ;NO WORKING SET LIST INDEX INPUT
62 00 58 09 E4 0013 359 BBSC #LCK_V_GLOBAL,R8,15$ ;ASSUME NOT A GLOBAL PAGE
01 00 0C 0017 360 15$: PROBER #0,#T,TR2) ;IS THIS PAGE ACCESSIBLE?
      04   12 001B 361 BNEQ 30$           ;BRANCH IF YES
50 0C 3C 001D 362 MOVZWL #SSS_ACCVIO,RO ;ACCESS VIOLATION IF NOT
      05   0020 363 RSB
      62   95 0021 364 30$: TSTB (R2)      ;FAULT THE PAGE INTO THE WORKING SET
      0023 365 :
      0023 366 : R0 = ACCESS MODE FOR PAGE OWNER CHECK
      0023 367 : R1 = 0 INDICATING NO INPUT WORKING SET LIST INDEX
      0023 368 : R2 = VIRTUAL ADDRESS
      0023 369 :
      0023 370 DSBINT #IPL$_SYNCH        ;RAISE TO SYNCH TO INTERLOCK PFN DATA
      0029 371 :
      07   BB 0029 372 PUSHR #^M<R0,R1,R2> ;PUSHING CALLERS IPL
50 FFD2' 30 002B 373 BSBW MMG$SVAPTECHK ;PUSH VA, LOCK/UNLOCK FLAG, ACCESS MODE
      63   D0 002E 374 MOVL (R3),R0      ;RETURN R3 = SYS VA OF PAGE TABLE ENTRY
      1A   19 0031 375 BLSS 50$          ;R0 = PAGE TABLE ENTRY
      0033 376 :
      0033 377 : PAGE NOT VALID!! MUST HAVE SWAPPED JUST AFTER TOUCHING IT, TRY AGAIN
      0033 378 :
      07   BA 0033 379 35$: POPR #^M<R0,R1,R2> ;SO WE CAN FAULT IT AGAIN
      E7   11 0035 380 ENBINT            ;TOUCH THE PAGE AGAIN
      0038 381 BRB 30$                   :
      003A 382 :
      50 01EC 8F 3C 003A 383 40$: MOVZWL #SSS_PAGOWNVIO,RO ;PAGE OWNER VIOLATION
      03   11 003F 384 BRB 44$          :
      50 0144 24 3C 0041 385 42$: MOVZWL #SSS_NOPRIV,RO ;NO PRIVILEGE
      0144 31 0044 386 44$: BRW 110$        :
      0047 387 :
      0047 388 : THIS PAGE IS NOW VALID AND IN THE WORKING SET
      0047 389 :
      0047 390 :
      50 09 3C 0047 391 47$: MOVZWL #SSS_WASSET,RO ;SUCCESSFUL RETURN PATH FOR SHARED
      013E 31 004A 392 BRW 110$          ;MEMORY AND PFN MAPPED PAGES
      004D 393 :
      8E 50 02 17 ED 004D 394 50$: CMPZV #PTE$V_OWN,#PTESS_OWN,RO,(SP)+ ;CHECK PAGE OWNERSHIP
      02   E6 19 0052 395 BLSS 40$          ;BRANCH IF OWNER VIOLATION
      0054 396 ULKGBLWSLE:
      EF 50 15 E0 0054 397 BBS #PTE$V_WINDOW,RO,47$ ;RETURN SUCCESS FOR PFN MAPPED PAGES
      50 50 15 00 EF 0058 398 EXTZV #PTE$V_PFN,#PTESS_PFN,RO,RO ;RO = PAGE FRAME NUMBER
      00000000'EF 50 D1 005D 399 CMPL RO_MMGSGL_MAXPFN ;CHECK FOR PAGE IN SHARED MEMORY
      E1 1A 0064 400 BGTRU 47$          ;RETURN SUCCESS FOR SUCH PAGES
      53 00C8 C5 C2 0066 401 SUBL PHDSL_P0BR(R5),R3 ;BYTE OFFSET OF PAGE TABLE ENTRY
      52 53 F7 8F 78 006B 402 ASHL #-9,R3,R2 ;BYTE INDEX OF CONTAINING PAGE TABLE

```

```

0070 403
0070 404 ASSUME PFNSC_PROCESS EQ 0
0070 405 ASSUME PFNSC_SYSTEM EQ 1
0070 406 ASSUME PFNSC_GLOBAL EQ 2
0070 407 ASSUME PFNSC_GBLWRT EQ 3
0070 408 ASSUME PFNSC_PPGTBL EQ 4
0070 409 ASSUME PFNSC_GPGTBL EQ 5
0070 410
0070 411 EXTV #PFNSV_PAGTYP,#PFN$S_PAGTYP,- ;SEE IF PAGE IS A TYPE
0073 412 @W^PFNSAB_TYPÉ[R0],RT ;THAT CAN BE LOCKED
0078 413 BEQL 56$ ;PROCESS PAGES MAY BE LOCKED
007A 414 BBSS #LCK_V_GLOBAL,R8,52$ ;PAGE IS GLOBAL (OR ERROR)
007E 415 52$: CMPL R1,#PFNSC_GLOBAL ;SEE IF PAGE IS GLOBAL OR GLOBAL WRITE
0081 416 BEQL 54$ ;BRANCH IF GLOBAL READ ONLY
0083 417 BLSS 42$ ;BRANCH IF PAGE TYPE NOT LOCKABLE
0085 418
0085 419 ; PAGE IS GLOBAL WRITE, MAY BE LOCKED IN MEMORY, BUT NOT IN WORKING SET BECAUSE
0085 420 ; THE SWAPPER DROPS GLOBAL WRITABLE PAGES FROM THE WORKING SET ON OUTSWAP
0085 421
0085 422 BBC #LCK_V_LCKPAG,R8,54$ ;OK IF UNLOCKING
0089 423 BBS #WSL$V_WSLOCK,R8,42$ ;CANNOT LOCK GLOBAL WRITE IN WORKING SET
008D 424 54$: MOVZWL (SP),RT ;NORMALLY 0, WSLX IF ULKGBLWSLE
0090 425 BNEQ 57$ ;BRANCH IF IT IS A WSLX
0092 426
0092 427 ; NEED TO SCAN FOR THE WORKING SET LIST INDEX SINCE THE WSLX ARRAY
0092 428 ; DOES NOT CONTAIN THE BACKPOINTER FOR GLOBAL PAGES
0092 429
52 08 AE 52 DD 0092 430 PUSHL R2 ;SAVE INDEX OF PAGE TABLE CONTAINING THIS PA
0157 30 0094 431 MOVL 8(SP),R2 ;RECOVER VIRTUAL ADDRESS
0098 432 BSBW MMG$SCNWSLX ;SCAN FOR WORKING SET LIST INDEX
0098 433
0098 434 ; MAY HAVE BEEN SWAPPED, RECALCULATE PTE ADR, REFETCH PFN
0098 435
50 00C8 D543 15 00 EF 0098 436 EXTZV #PTESV_PFN,#PTESS_PFN,- ;REFETCH PFN GIVEN THE
04 BA 009E 437 @PHDSI_@POB(R5)[R3],R6 ;BYTE OFFSET OF PTE IN PAGE TABLE
51 D5 00A3 438 POPR #^M<R25 ;RECOVER SAVE R2
35 12 00A5 439 TSTL R1 ;FOUND THE WORKING SET LIST INDEX?
35 12 00A7 440 BNEQ 57$ ;BRANCH IF YES
00A9 441
00A9 442 ; CHECK FOR A RESIDENT GLOBAL SECTION PAGE
00A9 443
51 0000'DF40 00 00A9 444 MOVL @W^PFNSAL_BAK[R0],R1 ;GET BACKING STORE
1F 51 16 E1 00AF 445 BBC #PTESV_TYP0,R1,55$ ;MUST BE PAGE-FILE
51 51 32 00B3 446 CVTWL R1,R1 ;ISOLATE GLOBAL SECTION TABLE INDEX
50 20 A0 CO 00B6 447 MOVAL @L^MMG$GL_SYSPHD,R0 ;ADDRESS OF SYSTEM HEADER
50 6041 DE 00BD 448 ADDL PHDSL_PSTBASOFF(R0),R0 ;GET SECTION TABLE BASE
03 0000'CO 00000000'8F E1 00C1 449 MOVAL (R0)[R1],R0 ;GET THE CORRECT SECTION TABLE ENTRY
FF75 31 00CF 450 BBC #SEC$V_RÉSIDENT,SEC$W_FLAGS(R0),55$ ;RETURN SUCCESS
00D2 451 BRW 47$ ;JOIN COMMON RETRY LOGIC
00D2 452
00D2 453 ; THE GLOBAL PAGE WAS REMOVED FROM THE WORKING SET. THE ROUTINE WILL BE
00D2 454 ; BACKED OUT AFTER THE STACK IS PUT INTO A CONSISTENT STATE.
00D2 455
FC AD 00D2 456 55$: PUSHL MMG$L_MAXACMODE(FP) ;PUSH ACCESS MODE FOR OWNER CHECK
FF5B 31 00D5 457 BRW 35$ ;JOIN COMMON RETRY LOGIC
00D8 458
00D8 459 ; PROCESS PAGE, GET WORKING SET LIST INDEX FROM PFN DATA BASE

```

00D8 460 :
 00D8 461 56\$: PFN REFERENCE -
 00D8 462 <@W^PFNSAx_WSLX[R0],R1>,- ;R1 = INDEX TO WORKING SET LIST ENTR
 00D8 463 LONG OPCODE=MOVL -
 00D8 464 IMAGE=SYS_NONPAGED
 53 6541 DE 00DE 465 57\$: MOVAL (R5)[R1],R3 ;R3 = ADDRESS OF ENTRY
 00E2 466 :
 00E2 467 R0 = PAGE FRAME NUMBER
 00E2 468 R1 = WORKING SET LIST INDEX FOR THIS PAGE
 00E2 469 R2 = INDEX OF PAGE TABLE CONTAINING THIS PAGE
 00E2 470 R3 = WORKING SET LIST ENTRY ADDRESS
 00E2 471 0(SP) = SCRATCH
 00E2 472 4(SP) = SAVED VIRTUAL ADDRESS
 00E2 473 :
 5B 58 08 E1 00E2 474 58\$: BBC #LCK_V_LCKPAG,R8,80\$;BRANCH IF UNLOCKING
 00E6 475 :
 00E6 476 LOCK THE PAGE IN THE WORKING SET LIST
 00E6 477 :
 63 58 93 00E6 478 BITB R8,(R3) :ALREADY LOCKED?
 07 13 00E9 479 BEQL 60\$: BRANCH IF NOT
 58 10 10 09 F0 00EB 480 INSV #SSS_WASSET,#16,#16,R8 PREVIOUS STATE OF LOCK WAS SET
 78 11 00FO 481 BRB 96\$: NOTE INSV CLEARS Z BIT
 CF 8F 8B 00F2 482 60\$: BICB3 #^C<WSLSM_WSLock ! WSLSM_PFNLOCK>,- ;SAVE THE CURRENT STATE
 6E 63 00F5 483 (R3),(SP) OF THE LOCK BITS
 0E 12 00F7 484 BNEQ 62\$: BRANCH IF NEITHER IS SET
 76 A5 01 B1 00F9 485 CMPW #1_PHDSW_EXTDYNWS(R5) POSSIBLE TO LOCK 2 MORE PAGES?
 08 1F 00FD 486 BLSSU 62\$: BRANCH IF YES
 50 0194 8F 3C 00FF 487 MOVZWL #SSS_LKSETFUL,RO LOCKED PORTION OF WORKING SET FULL
 0084 31 0104 488 BRW 110\$:
 15 58 04 E1 0107 489 62\$: BBC #WSLSV_PFNLOCK,R8,68\$ BRANCH IF LOCKING IN WORKING SET
 0B 58 09 E1 010B 490 BBC #LCK_V_GLOBAL,R8,64\$ BRANCH IF PAGE IS NOT GLOBAL
 010F 491 PFN_REFERENCE -
 010F 492 ACBW <#1,#1,@W^PFNSAx_WSLX[R0],64\$>,- ;BRANCH IF FIRST LOCK IN
 010F 493 LONG OPCODE=ACBL -
 010F 494 IMAGE=SYS_NONPAGED
 0118 495 :MEMORY OF THIS GLOBAL PAGE
 06 11 0118 496 BRB 68\$:
 0000'CF D7 011A 497 64\$: DECL W^MMGSGL_PFNLOCK :CAN ANY MORE PAGES BE LOCKED IN MEMORY?
 71 19 011E 498 BLSS 150\$: BRANCH IF NO, TOO MANY ALREADY LOCKED
 63 58 88 0120 499 68\$: BISB R8,(R3) :SET NEW LOCK BIT IN WSLE
 6E 95 0123 500 TSTB (SP) :WAS ENTRY ALREADY IN LOCKED PART OF WS?
 5F 12 0125 501 BNEQ 100\$: BRANCH IF YES
 0127 502 :
 0127 503 ENTRY MUST BE PLACED IN THE LOCKED PORTION OF THE WORKING SET
 0127 504 :
 7C 10 0127 505 BSBB SWAPLOCK :SWAP THE WORKING SET LIST ENTRIES
 62 96 0129 506 INCB (R2) :COUNT ANOTHER LOCKED WSLE FOR THIS PAGE TAB
 03 14 012B 507 BGTR 70\$: BRANCH IF NOT THE FIRST IN THIS PT
 6C A5 B6 012D 508 INCW PHDSW_PTCNTLCK(R5) :ANOTHER PAGE TABLE WITH LOCKED WSLE'S
 OE A5 B6 0130 509 70\$: INCW PHDSW_WSDYN(R5) :DYNAMIC PART OF LIST
 0133 510 : STARTS ONE ENTRY HIGHER
 OE A5 10 A5 B1 0133 511 CMPW PHDSW_WSNEXT(R5),PHDSW_WSDYN(R5) :IS WSNEXT NO LONGER
 0138 512 : POINTING INTO DYNAMIC REGION?
 10 A5 12 A5 B0 0138 513 BGEQU 98\$: BRANCH IF IT'S OK
 42 11 013A 514 MOVW PHDSW_WSLAST(R5),PHDSW_WSNEXT(R5) :OTHERWISE POINT IT TO END
 013F 515 BRB 98\$: EXIT SUCCESSFULLY
 0141 516 :

0141 517 : UNLOCK PAGE FROM WORKING SET LIST

63 58 93 0141 0141 518 :
 07 01 F0 0144 0144 519 80\$: BITB R8,(R3) ;IS THIS PAGE LOCKED?
 58 10 10 01 39 11 0146 0146 520 BNEQ 90\$;BRANCH IF YES
 13 63 04 E1 014D 014B 521 INSV #SSS_WASCLR,#16,#16,R8 ;AT LEAST ONE PAGE WAS ALREADY UNLOCKED
 OF 58 04 E1 0151 0151 522 BRB 100\$
 07 58 09 E1 0155 0155 523 90\$: BBC #WSLSV_PFNLOCK,(R3),94\$;BRANCH IF PAGE NOT LOCKED IN MEMORY
 04 04 E1 0159 0159 524 BBC #WSLSV_PFNLOCK,R8,94\$;BRANCH IF NOT UNLOCKING IT FROM MEMORY
 07 58 09 E1 0159 0159 525 BBC #LCK_V_GLOBAL,R8,92\$;BRANCH IF NOT A GLOBAL PAGE
 0159 0159 526 PFN_REFERENCE -
 0159 0159 527 DECW <@W^PFNSAX_WSLX[R0]>,- ;ONE LESS MEMORY LOCK FOR THIS GLOBA
 0159 0159 528 LONG_OPCODE=DECL -
 04 14 015E 0160 530 BGTR 94\$;BRANCH IF NOT LAST MEMORY LOCK
 0000'CF D6 0160 531 92\$: INCL W^MMGSGL_PFNLOCK ;ANOTHER PAGE MAY BE LOCKED IN MEMORY
 63 58 8A 0164 0164 532 94\$: BICB R8,(R3) ;CLEAR THE DESIRED LOCK BIT(S)
 63 30 93 0167 0167 533 BITB #<WSLSM_WSLOCK ! WSLSM_PFNLOCK>,(R3) ;MUST PAGE REMAIN IN THE
 016A 016A 534 016A 016A 535 96\$: BNEQ 100\$;LOCKED PORTION OF THE WORKING SET?
 1A 12 016A 016C 536 ;BRANCH IF YES, IT IS STILL LOCKED
 016C 537 : MOVE THIS WORKING SET LIST ENTRY OUT OF THE LOCKED PORTION OF THE WORKING ET
 016C 538 :
 0E A5 B7 016C 539 DECW PHDSW_WSDYN(R5) ;INCLUDE LAST LOCKED SLOT IN DYNAMIC REGION
 34 10 016F 540 BSBB SWAPLOCK ;SWAP ENTRIES WITH THE LAST LOCKED ONE
 52 DD 0171 541 PUSHL R2 ;SAVE BYTE INDEX OF PAGE TABLE PAGE
 52 FE86' 0173 542 MOVL 8(SP),R2 ;RESTORE VA
 52 BED0 0177 543 BSBW MMG\$SVAPTECHK ;GET R3 = SVAPTE
 50 01 CE 017D 544 POPL R2 ;RESTORE BYTE INDEX OF PAGE TABLE PAGE
 50 FE7D' 0180 545 MNEG L #1,R0 ;INDICATOR DECREMENT OF PTWSLELOCK ARRAY
 50 FE7A' 0183 546 BSBW MMG\$MOVPTLOCK1 ;ONE LESS REASON PT PAGE MUST BE LOCKED
 50 58 F0 8F 78 0186 547 98\$: BSBW MMG\$EXTRADYNWS ;RECALCULATE EXTRA DYNAMIC WSLE COUNT
 06 BA 0188 548 100\$: ASHL #16,R8,R0 ;RETURN WASSET OR WASCLR
 018D 549 110\$: POPR #^M<R1,R2> ;RESTORE R2 = VIRTUAL ADDRESS
 05 0190 550 ENBINT ;AND RESTORE THE CALLING IPL
 0191 551 RSB ;AND RETURN
 0191 552 :
 0191 553 : TOO MANY PAGES LOCKED IN MEMORY TO ALLOW ANOTHER TO BE LOCKED
 0191 554 :
 05 58 09 E1 0191 555 150\$: BBC #LCK_V_GLOBAL,R8,155\$;BRANCH IF PAGE NOT GLOBAL
 0195 556 PFN_REFERENCE -
 0195 557 DECW <@W^PFNSAX_WSLX[R0]>,- ;THIS GLOBAL PAGE NOT LOCKED IN MEMO
 0195 558 LONG_OPCODE=DECL,-
 0195 559 IMAGE=SYS_NONPAGED
 0195 560 155\$: INCL W^MMGSGL_PFNLOCK ;FIX THE COUNTER
 50 0000'CF D6 019A 561 MOVZWL #SSS_LCKPAGFUL,RC ;NO MORE PAGES MAY BE LOCKED IN MEMORY
 E6 8F 3C 019E 562 BRB 110\$
 E6 11 01A3 563 .DSABL LSB

01A5 567 .SBTTL SWAPLOCK - INTERFACE TO SWAPWSLE FROM LOCK/UNLOCK
01A5 568
01A5 569
01A5 570 : INPUTS:
01A5 571
01A5 572 R0 = PFN OF THE FIRST WORKING SET LIST SLOT
01A5 573 R1 = INDEX TO FIRST WORKING SET LIST SLOT (WSLX1)
01A5 574 R2 = BYTE INDEX OF PAGE TABLE CONTAINING THE PAGE BEING LOCKED/UNLOCKED
01A5 575 R4 = PCB ADDRESS
01A5 576 R5 = PROCESS HEADER ADDRESS - P1 OR SYSTEM SPACE
01A5 577 IPL = SYNCH, ESSENTIAL FOR MANIPULATING WORKING SET LIST
01A5 578 FOR SYSTEM SPACE, AND BECAUSE PFN DATA BASE IS CHANGED
01A5 579
01A5 580 : OUTPUTS:
01A5 581
01A5 582 R0 PRESERVED
01A5 583 R2 = ADDRESS OF BYTE CONTAINING COUNT OF LOCKED WORKING SET LIST
01A5 584 ENTRIES IN THE PAGE TABLE
01A5 585 SPECIFIED WORKING SET LIST ENTRIES ARE SWAPPED
01A5 586
01A5 587 SWAPLOCK:
52 0E A5 52 DD 01A5 588 PUSHL R2 :SAVE PAGE TABLE NUMBER
52 09 10 3C 01A7 589 MOVZWL PHDSW_WSDYN(R5),R2 :INDEX TO SECOND WORKING SET LIST SLOT (WSLX
52 64 A5 C1 01AB 590 BSBB MMG\$SWAPWSLE :SWAP THE WORKING SET LIST ENTRIES
52 8E C0 01B2 01AD 591 ADDL3 PHDSL_PTWSLELCK(R5),R5,R2 :BASE ADDRESS OF ARRAY OF COUNTS OF
05 01B5 592 ADDL (SP)+,R2 :LOCKED WORKING SET LIST ENTRIES
05 01B5 593 RSB :ADDRESS OF COUNT BYTE FOR # OF LOCKED
05 01B5 594 WSLE'S IN THE PAGE TABLE
05 01B5 595

01B6 597 .SBTTL SWAPWSLE - SWAP WORKING SET LIST ENTRIES
01B6 598 :++
01B6 599 : FUNCTIONAL DESCRIPTION:
01B6 600
01B6 601 THIS ROUTINE SWAPS THE CONTENTS OF THE FIRST WORKING SET LIST SLOT
01B6 602 WHICH MUST BE ACTIVE WITH THAT OF THE SECOND WORKING SET LIST SLOT
01B6 603 WHICH MAY BE EITHER ACTIVE OR FREE.
01B6 604
01B6 605 : CALLING SEQUENCE:
01B6 606
01B6 607 BSBW MMG\$SWAPWSLE
01B6 608
01B6 609 : INPUT PARAMETERS:
01B6 610
01B6 611 R0 = PFN OF THE FIRST WORKING SET LIST SLOT
01B6 612 R1 = INDEX TO FIRST WORKING SET LIST SLOT (WSLX1)
01B6 613 R2 = INDEX TO SECOND WORKING SET LIST SLOT (WSLX2)
01B6 614 R4 = PCB ADDRESS
01B6 615 R5 = PROCESS HEADER ADDRESS - P1 OR SYSTEM SPACE
01B6 616 IPL = SYNCH, ESSENTIAL FOR MANIPULATING WORKING SET LIST
01B6 617 FOR SYSTEM SPACE, AND BECAUSE PFN DATA BASE IS CHANGED
01B6 618
01B6 619 : IMPLICIT INPUTS:
01B6 620
01B6 621
01B6 622
01B6 623 : OUTPUT PARAMETERS:
01B6 624
01B6 625 R0 PRESERVED
01B6 626
01B6 627 : IMPLICIT OUTPUTS:
01B6 628
01B6 629
01B6 630
01B6 631 : COMPLETION CODES:
01B6 632
01B6 633
01B6 634
01B6 635 : SIDE EFFECTS:
01B6 636
01B6 637
01B6 638
01B6 639 :--

```

01B6 641 : ****
01B6 642 : **** THE FOLLOWING CODE MUST BE RESIDENT ****
01B6 643 :
01B6 644 :
01B6 645 :
000001B6 646 .PSECT $MMGCOD
01B6 647 :
01B6 648 :
01B6 649 :
01B6 650 .ENABL I,SB
01B6 651 :
01B6 652 MMG$SWAPWSLE:::
52 51 D1 01B6 653 CMPL R1,R2 ;IS THE SWAP A NOP?
32 13 01B9 654 BEQL 90$ ;BRANCH IF YES
6541 DD 01BB 655 PUSHL (R5)[R1] ;PUSH WSLE1
05 BB 01BE 656 PUSHR #^M<R0,R2> ;PUSH WSLX2, PFN(WSLX1)
52 6542 D0 01C0 657 MOVL (R5)[R2],R2 ;GET WSLE2
05 12 01C4 658 BNEQ 50$ ;BRANCH IF NOT AN EMPTY ENTRY
6541 D4 01C6 659 CLRL (R5)[R1] ;JUST FREE WSLX1
0F 11 01C9 660 BRB 70$ ;AND GO SET UP WSLX2
20 52 E9 01CB 661 50$: BLBC R2,SWAPWSLEBUG ;BRANCH IF WSLE NOT VALID, ERROR
51 DD 01CE 662 PUSHL R1 ;SAVE WSLX1
FE2D' 30 01D0 663 BSBW MMG$SVAPTECHK ;RETURN R3 = SYS VA OF PAGE TABLE ENTRY
FE2A' 30 01D3 664 BSBW MMG$WSLEPFN ;GET PFN FROM SVAPTE OF THIS WSLE
02 BA 01D6 665 POPR #^M<R1> ;R1 = WSLX1
02 10 01D8 666 BSBB SETWSLE ;STORE WSLE AND WSLX
01DA 667 :
01DA 668 : MOVE THE OLD WSLE1 INTO SLOT INDEXED BY WSLX2
01DA 669 : 0(SP) = PFN OF WSLE1, 4(SP) = WSLX2, 8(SP) = WSLE1
01DA 670 :
07 BA 01DA 671 70$: POPR #^M<R0,R1,R2> ;R0=PFN, R1=WSLX2, R2=WSLE1
01DC 672 :
01DC 673 : STORE WORKING SET LIST ENTRY INTO ITS WORKING SET LIST ENTRY SLOT
01DC 674 : AND STORE WSLX INTO PFN DATA BASE IF PAGE TYPE IS NOT GLOBAL
01DC 675 : R0 = PAGE FRAME NUMBER
01DC 676 : R1 = WSLX, R2 = WORKING SET LIST ENTRY WITH PAGE TYPE
01DC 677 :
01DC 678 SETWSLE:
02 52 6541 52 D0 01DC 679 MOVL R2,(R5)[R1] ;STORE WORKING SET LIST ENTRY
01 03 01 EC 01E0 680 CMPV #WSL$V_PAGTYP,#WSL$S_PAGTYP,R2,#PFN$C_GLOBAL ;GLOBAL PAGE?
06 18 01E5 681 BGEQ 90$ ;BRANCH IF NOT
01E7 682 PFN_REFERENCE - -
01E7 683 MOVW <R1,>W^PFNSAx_WSLX[R0]>,- ;SET PFN WSLX
01E7 684 LONG_OPCODE=MOVL,-
01E7 685 IMAGE=SYS_NONPAGED
05 01ED 686 90$: RSB ;AND RETURN
01EE 687 :
01EE 688 : LENGTH VIOLATION FOR VIRTUAL ADDRESS IN WORKING SET LIST
01EE 689 :
01EE 690 SWAPWSLEBUG:
01EE 691 BUG_CHECK SWAPWSLE,FATAL
01F2 692 :
01F2 693 .DSABL LSB

```

01F2 695 .SBTTL SCNWSLX - SCAN FOR WORKING SET LIST INDEX
01F2 696 ++
01F2 697
01F2 698 FUNCTIONAL DESCRIPTION:
01F2 699
01F2 700 THIS ROUTINE SCANS THE ENTIRE WORKING SET FOR THE SPECIFIED
01F2 701 VIRTUAL ADDRESS. IT IS CALLED AT IPL=SYNCH, LOWERS TO IPL=ASTDEL
01F2 702 TO DO THE SCAN AND RAISES BACK TO SYNCH WHEN COMPLETED.
01F2 703 DROPPING IPL TO ASTDEL AND THUS MAKING THE PROCESS ELIGIBLE
01F2 704 FOR SWAPPING, PLACES THE FOLLOWING RESTRICTIONS ON THIS CODE AND
01F2 705 ITS CALLERS.
01F2 706
01F2 707 1) THIS CODE MAY NOT FAULT PROCESS PAGES SINCE THAT MIGHT
01F2 708 FAULT OUT THE DESIRED WORKING SET LIST ENTRY.
01F2 709 2) THE CALLING CODE MAY NOT KEEP ANY SYSTEM SPACE ADDRESSES
01F2 710 FOR THE PROCESS HEADER OR PAGE TABLE ENTRIES SINCE THE HEADER BASE
01F2 711 ADDRESS MAY CHANGE IF THE PROCESS IS SWAPPED.
01F2 712 3) THE CALLING CODE MAY NOT HOLD A PFN FOR A GIVEN PAGE SINCE
01F2 713 THAT TOO COULD CHANGE IF THE PROCESS WERE SWAPPED.
01F2 714
01F2 715 CALLING SEQUENCE:
01F2 716
01F2 717 BSBW MMG\$SCNWSLX
01F2 718
01F2 719 INPUTS:
01F2 720
01F2 721 R2 = VIRTUAL ADDRESS TO SCAN FOR
01F2 722 R5 = P1 ADDRESS OF PROCESS HEADER
01F2 723 IPL = SYNCH, DROPS TO ASTDEL, RESTORES TO SYNCH ON RETURN
01F2 724 SEE FUNCTIONAL DESCRIPTION ABOVE!
01F2 725
01F2 726 OUTPUTS:
01F2 727
01F2 728 R1 = WORKING SET LIST INDEX IF FOUND
01F2 729 = 0 IF NOT FOUND
01F2 730 R3 = PRESERVED
01F2 731 IPL = IS LOWERED TO ASTDEL AND RESTORED TO SYNCH ON RETURN
01F2 732 SEE FUNCTIONAL DESCRIPTION ABOVE!
01F2 733 ;--

```

01F2 735 :
01F2 736 : ****
01F2 737 : **** THE FOLLOWING CODE MUST BE RESIDENT ****
01F2 738 : ****
01F2 739 : ****
000001F2 740 .PSECT $MMGCOD
01F2 741 :
01F2 742 :
01F2 743 :
01F2 744 MMGSSCNWSLX:::
52 52 F7 8F 78 01F5 745 SETIPL #IPL$_ASTDEL ; LOWER IPL FROM SYNCH TO ASTDEL
51 0C A5 3C 01FA 746 ASHL #-VASS BYTE,R2,R2 ; JUST INTERESTED IN PAGE NUMBER BITS
51 6541 DE 01FE 747 MOVZWL PHDSW $SLLOCK(R5),R1 ; INDEX TO FIRST NON-PERMANENT WSLE
50 12 A5 3C 0202 748 MOVAL (R5)[R1],R1 ; ADDRESS OF FIRST WSLE
50 6540 DE 0206 749 MOVZWL PHDSW WSLAST(R5),R0 ; INDEX TO LAST (INCLUSIVE) WSLE
52 61 17 09 EC 020A 750 MOVAL (R5)[R0],R0 ; ADDRESS OF LAST WSLE
      0C 13 020F 751 20$: CMPV #VASV_VPG,#VASS_VPG,(R1),R2 ; IS THIS THE DESIRED WSLE?
      FFF3 51 04 50 F1 0211 752 BEQL 40$ ; BRANCH IF YES
      0217 753 ACBL R0,#4,R1,20$ ; LOOP THROUGH ALL THE SLOTS
      0217 754 :
      0217 755 : WORKING SET LIST ENTRY NOT FOUND FOR SPECIFIED VIRTUAL ADDRESS
      0217 756 :
51 D4 0217 757 30$: CLRL R1 ; RETURN IMPOSSIBLE WSLX
      0219 758 SETIPL #IPL$_SYNCH ; RESTORE IPL SYNCH
      05 021C 759 RSB ; AND RETURN TO CALLER
      021D 760 :
      021D 761 : FOUND THE WORKING SET LIST ENTRY
      021D 762 :
      021D 763 40$: SETIPL #IPL$_SYNCH ; BACK TO IPL SYNCH
      F4 61 E9 0220 764 BLBC (R1),30$ ; MAKE SURE SWAPPER DIDN'T JUST
      0223 765          ; RIP THE ENTRY OUT FROM UNDER US
      51 51 55 C2 0223 766 SUBL R5,R1 ; CONVERT ADDRESS
      1E 9C 0226 767 ROTL #<32-2>,R1,R1 ; BACK TO INDEX
      05 022A 768 RSB ; RETURN WITH WSLX IN R1

```

022B 770 .SBTTL MMG\$EXPKSTACK - EXPAND THE KERNEL STACK
022B 771 :++
022B 772 :
022B 773 : FUNCTIONAL DESCRIPTION:
022B 774 :
022B 775 : THIS ROUTINE EXPANDS THE KERNEL STACK BY ALLOCATING PHYSICAL MEMORY AND
022B 776 : USING THE RESERVED WORKING SET LIST ENTRIES. THIS ROUTINE RETURNS WITH
022B 777 : SUCCESS IF THE STACK HAS ALREADY BEEN EXPANDED.
022B 778 :
022B 779 : CALLING SEQUENCE:
022B 780 :
022B 781 : CALLS #0,MMG\$EXPKSTK
022B 782 :
022B 783 : INPUTS:
022B 784 :
022B 785 : NONE
022B 786 :
022B 787 : OUTPUTS:
022B 788 :
022B 789 : NONE
022B 790 :--

```

022B 792 : ****
022B 793 : **** THE FOLLOWING CODE MUST BE RESIDENT ****
022B 794 :
022B 795 : **** THE FOLLOWING CODE MUST BE RESIDENT ****
022B 796 : ****
0000022B 797 : .PSECT $MMGCOD
022B 798 :
022B 799 : ****
022B 800 :
022B 801 MMGSEXPKSTK:: WORD ^M<R2,R3,R4,R5,R6>
007C 022B 802 DSBINT #IPLS ASTDEL ;DON'T LET THINGS GET CONFUSED
      54 0000'CF  DO 0233 803 MOVL W$CH$GL CURPCB,R4 ;R4 = PCB
      00000000'EF  DO 0238 804 MOVL CTL$GL_PHD,R5 ;R5 = P1 PHD
      FFFFFFFF'8F  DO 023F 805 MOVL #SWP$C_KSTACK_EX WSL- ;WSL INDEX FOR FIRST EXPANSION PAGE
      56 0245 806 +SWP$C_KSTACK_EX=1,R6
      6546 D5 0246 807 TSTL (R5)[R6]
      7B 12 0249 808 BNEQ 30S ;HAVE WE ALREADY EXPANDED
      52 00000000'8F  DO 024B 810 MOVL #CTL$GL_KSTKBASEXP,R2 ;YES
      FDA8' 30 0255 811 10$: SETIPL #IPLS SYNCH ;VA OF FIRST EXPANSION PAGE
      7E 52 7D 0258 812 BSBW MMG$SVAPTECHK ;GET SVAPTE IN R3
      FDA2' 30 025B 813 MOVQ R2,-(SP)
      52 6E 7D 025E 814 BSBW MMG$FREWSLE ;MAKE SURE WE CAN ADD A WSL PAGE
      69 50 E9 0261 815 MOVQ (SP) R2
      7E 10 A5 B0 0264 816 BLBC R0,100$ ;RO,100$?
      10 A5 56 B0 0268 817 MOVW PHDSW_WSNEXT(R5),-(SP) ;SAVE THE FREE POINTER
      FD91' 30 026C 818 MOVW R6,PHDSW_WSNEXT(R5) ;USE THIS WSL ENTRY
      10 A5 8E B0 026F 819 BSBW MMGSININEWPFN ;GET A PFN AND FILL IN WSL
      52 8E 7D 0273 820 MOVW (SP)+,PHDSW_WSNEXT(R5) ;RESTORE THE FREE POINTER
      50 D5 0276 821 MOVQ (SP)+,R2
      5E 19 0278 822 TSTL R0
      00000000'FF40 07 88 027A 823 BLSS 110$ ;NO PFN AVAILABLE
      D0000000 8F C9 0282 824 BISB #PFNSC_ACTIVE,0L^PFNSAB STATE[R0] ;MARK IT ACTIVE
      63 50 0288 825 BISL3 #<PTESM_VALID ! PTESC_SRKW>,- ;A VALID PTE
      6546 20 C8 028D 826 INVALID R2
      52 0200 C2 9E 0294 827 BISL #WSLSM_WSLOCK,(R5)[R6] ;SET LOCK BIT
      08 A5 B7 0291 828 DECW PHDSW_WSLIST(R5) ;WSL STARTS ONE PAGE SOONER
      512(R2),R2 ;NEXT PAGE
      ACBB #SWP$C_KSTACK_EX WSL,-1,R6,10$ ;NEXT WSL FOR EXPANSION
      00000000'EF 00000000'8F 9D 029C 832 MOVL #CTL$GE_KSTKBASEXP,CTL$AL_STACK-4 ;NEW STACK LIMITS
      00000000'8F 00000000'8F DO 02A4 833 MOVL #CTL$GL_KSTKBASEXP,CTL$AL_STACKLIM
      0000'8F 00 6E 00 2C 02BA 834 MOVC5 #0,(SP),#0,#^X200*SWP$C_KSTACK_EX,- ;ZERO THE NEW PAGES
      00000000'EF 02C1 835 CTL$GL_KSTKBASEXP
      50 01 3C 02C9 836 30$: ENBINT
      04 02CC 837 MOVZWL #SSS_NORMAL,RO
      02CD 838 RET
      FD30' 30 02CD 841 100$: BSBW MMG$RESRCWAIT ;PUT PROCESS ON CORRECT QUEUE
      7E DC 02D0 842 105$: MOVPSL -(SP) ;WAIT AT THIS PSL
      FD2B' 30 02D2 843 BSBW MMG$SVPCTX ;WAIT FOR RESOURCE
      FF7A 31 02D5 844 BRW 10$ ;TRY AGAIN
      50 0000'CF 7E 02D8 845 10$: MOVAQ W$CHSGQ_FPGWQ,RO ;FREE PAGE QUEUE
      08 A0 B6 02DD 846 INCW WQH$W_WQCNT(R0),R0 ;ONE MORE WAITER
      60 64 0E 02E0 847 INSQUE (R4),R0 ;QUEUE THE PCB
      60 64 0E 02E0 848

```

SYSLKWSET
V04-000

B 5
- LOCK/UNLOCK FROM WORKING SET/MEMORY
MMG\$EXPSTACK - EXPAND THE KERNEL STACK 16-SEP-1984 02:21:29 VAX/VMS Macro V04-00
5-SEP-1984 03:54:52 [SYS.SRC]SYSLKWSET.MAR;1

Page 20
(13)

2C A4 0A A0 B0 02E3 849
E6 11 02E8 850
02EA 851
02EA 852

MOVW WQH\$W_WQSTATE(R0),PCBSW_STATE(R4) ;SET WAIT STATE IN PCB
BRB 105\$
.END

SY
VO

PFN	= 000001E7 R 04	PFN\$C_PPGTBL	= 00000004
ACMODE	= 0000000C	PFN\$C_PROCESS	= 00000000
BIT...	= 0000000A	PFN\$C_SYSTEM	= 00000001
BUGS_SWAPWSLE	***** X 04	PFN\$S_PAGTYP	= 00000003
CTL\$AL_STACK	***** X 04	PHDSL_POBR	= 00000000
CTL\$AL_STACKLIM	***** X 04	PHDSL_PSTBASOFF	= 00000020
CTL\$GL_KSTKBASEXP	***** X 04	PHDSL_PTWSLELOCK	= 00000064
CTL\$GL_PHD	***** X 03	PHD\$W_EXTDYNWS	= 00000076
EXE\$LCRPAG	0000001E RG 03	PHD\$W_PTCNTLCK	= 0000006C
EXE\$LKSET	00000017 RG 03	PHD\$W_WSDYN	= 0000000E
EXE\$ULKPAG	00000007 RG 03	PHD\$W_WSLAST	= 00000012
EXE\$ULWSET	00000000 RG 03	PHD\$W_WSLIST	= 00000008
INADR	= 00000004	PHD\$W_WSLOCK	= 0000000C
IPLS_ASTDEL	= 00000002	PHD\$W_WSNEXT	= 00000010
IPLS_SYNCH	= 00000008	PHD\$W_WSQUOTA	= 00000018
LCK_M_GLOBAL	= 00000200	PHD\$W_WSSIZE	= 00000050
LCK_M_LCKPAG	= 00000100	PRS_IPL	= 00000012
LCK_V_GLOBAL	= 00000009	PRS_TBIS	= 0000003A
LCK_V_LCKPAG	= 00000008	PRV\$V_PSWAPM	= 0000000C
MMG\$CREDEL	***** X 03	PTE\$C_SRKW	= 50000000
MMG\$C_LENGTH	= FFFFFFFE4	PTE\$M_VALID	= 80000000
MMG\$ERPKSTK	0000022B RG 04	PTE\$S OWN	= 00000002
MMG\$EXTRADYNWS	***** X 04	PTE\$V_PFN	= 00000015
MMG\$FREWSLE	***** X 04	PTE\$V_PFN	= 00000017
MMG\$GL_MAXPFN	***** X 04	PTE\$V_TYPO	= 00000000
MMG\$GL_PFNLOCK	00000000 RG 02	PTE\$V_WINDOW	= 00000016
MMG\$GL_SYSPHD	***** X 04	RETADR	= 00000015
MMG\$INADRINI	***** X 03	SCH\$GL_CURPCB	= 00000008
MMG\$ININEWPFN	***** X 04	SCH\$GQ_FPGWQ	***** X 04
MMG\$LKULKULKULK	00000011 RG 04	SEC\$V_RESIDENT	***** X 04
MMG\$L_MAXACMODE	= FFFFFFFF C	SEC\$W_FLAGS	***** X 04
MMG\$M_OVPTLOCK1	***** X 04	SETWSCE	000001DC R 04
MMG\$RESRCWAIT	***** X 04	SIZ...	= 00000001
MMG\$RETRANGE	***** X 03	SSS_ACCVIO	= 0000000C
MMG\$SCNWSLX	000001F2 RG 04	SSS_LCKPAGFUL	= 000000D4
MMG\$SVAPTECHK	***** X 04	SSS_LKWSETFUL	= 00000194
MMG\$VPCTX	***** X 04	SSS_NOPRIV	= 00000024
MMG\$SWAPWSLE	000001B6 RG 04	SSS_NORMAL	= 00000001
MMG\$ULKGBLWSLE	00000000 RG 04	SSS_PAGOWNVIO	= 000001EC
MMG\$WSLEPFN	***** X 04	SSS_WASCLR	= 00000001
OPS_ACBL	= 000000F1	SSS_WASSET	= 00000009
OPS_ACBW	= 0000003D	SWAPLOCK	000001A5 R 04
OPSDECL	= 000000D7	SWAPWSLEBUG	000001EE R 04
OPSDECW	= 000000B7	SWP\$C_KSTACK_EX	***** X 04
OPSMOVL	= 000000D0	SWP\$C_KSTACK_EX_WSL	***** X 04
OPSMOVW	= 000000B0	SYSSADJWSL	***** X 03
OPSMOVZWL	= 0000003C	ULKGBLWSLE	00000054 R 04
PCBSQ_PRIV	= 00000084	VASS_BYTE	= 00000009
PCBSW_STATE	= 0000002C	VASS_VPG	= 00000017
PFNSAB_STATE	***** X 04	VASV_VPG	= 00000009
PFNSAB_TYPE	***** X 04	WQHSQ_WQCNT	= 00000008
PFNSAL_BAK	***** X 04	WQHSW_WQSTATE	= 0000000A
PFNSAX_WSLX	***** X 04	WSLSM_PFNLOCK	= 00000010
PFNSC_ACTIVE	= 00000007	WSLSM_WSLOCK	= 00000020
PFNSC_GBLWRT	= 00000003	WSL\$S_PAGTYP	= 00000003
PFNSC_GLOBAL	= 00000002		
PFNSC_GPGTBL	= 00000005		

SYSLKWSET Symbol table

= LOCK/UNLOCK FROM WORKING SET/MEMORY

16-SEP-1984 02:21:29 VAX/VMS Macro V04-00
5-SEP-1984 03:54:52 [SYS.SRC]SYSLWKSET.MAR;1

Page 22
(13)

WSL\$V_PAGTYP
WSL\$V_PFNLOCK
WSL\$V_WSLOCK

= 00000001
= 00000004
= 00000005

+-----+
! Psect synopsis !
+-----+

PSECT name

Allocation PSECT No. Attributes

```
    .ABS .
$ABSS
$$$210
Y$EXEPAGED
$MMGCOD
Z$INIT$PFN_FIXUP_TABLE
```

00000000	(0.)	00	(0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
00000000	(0.)	01	(1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NCVEC	BYTE
00000004	(4.)	02	(2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	LONG
000000B1	(177.)	03	(3.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
000002EA	(746.)	04	(4.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
0000001E	(30.)	05	(5.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE

! Performance indicators !

Phase

Page faults CPU Time Elapsed Time

Initialization
Command processing
Pass 1
Symbol table sort
Pass 2
Symbol table output
Psect synopsis output
Cross-reference output
Assembler run total

The working set limit was 1500 pages.

80740 bytes (158 pages) of virtual memory were used to buffer the intermediate code.

There were 70 pages of symbol table space allocated to hold 1392 non-local and 49 local symbols.

852 source lines were read in Pass 1, producing 34 object records in Pass 2.

31 pages of virtual memory were used to define 29 macros.

+-----+
! Macro library statistics !
+-----+

Macro Library name

Macros defined

\$255\$DUA28:[SYS.OBJ]LIB.MLB;1
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2
TOTALS (all libraries)

17

1553 GET\$ were required to define 26 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:SYSLKWSET/OBJ=OBJ\$:SYSLKWSET MSRC\$:SYSLKWSET/UPDATE=(ENH\$:SYSLKWSET)+EXECML\$/LIB

0386 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SYSARAM
LIS

SYSMTRACE
LIS

SYSIMGSTA
LIS

SYSLHM
LIS

SYSLOAVEC
LIS

SYSLKSET
LIS

SYSMAILBX
LIS